This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.-2. (Canceled).

3. (Currently Amended) A method of optimizing a database query, the method comprising:

determining a retrieval status for a resource used by the database query, wherein determining the retrieval status includes determining a percentage of the resource that is resident in a working memory; ~~and~~

generating an access plan for the database query using the determined retrieval status for the resource, wherein generating the access plan includes:

generating a plurality of alternate access plans;

calculating a cost for each alternate access plan using the determined retrieval status for the resource; and

selecting one of the alternate access plans based upon the calculated costs for each alternate access plan;

storing the access plan, including associating with the stored access plan a retrieval status assumption for the stored access plan, the retrieval status assumption representing the determined retrieval status used to generate the access plan; and

in response to a request to execute the stored access plan:

determining a current retrieval status for the resource, wherein determining the current retrieval status includes determining a percentage of the resource that is currently resident in the working memory;

comparing the current retrieval status with the retrieval status assumption; and

selectively generating another access plan for the database query using the current retrieval status based upon the comparison of the current retrieval status with the retrieval status assumption.

4. (Original) The method of claim 3, wherein the resource is selected from the group consisting of a database file, a database table, an index, a temporary result set, a temporary file, a hash table, and combinations thereof.

5. (Original) The method of claim 3, wherein determining the retrieval status includes determining whether at least a portion of the resource is resident in working memory.

6. (Original) The method of claim 3, wherein determining the retrieval status includes determining whether at least a portion of the resource is resident in a cache memory.

7. (Original) The method of claim 3, wherein determining the retrieval status includes determining whether at least a portion of the resource is resident in a local or a remote memory.

8. (Canceled).

9. (Original) The method of claim 3, wherein determining the retrieval status includes accessing a resource manager to obtain the retrieval status of the resource.

10. (Currently Amended) The method of claim 9, further comprising tracking, with the resource manager, a the percentage of the resource that is in the working memory.

11. (Original) The method of claim 10, further comprising storing the percentage of the resource that is in working memory in a header for a persistent copy of the resource.

12. (Canceled).

13. (Currently Amended) The method of claim 3 ~~12, wherein the retrieval status for the resource indicates a percentage of the resource that is resident in working memory,~~ and wherein calculating the cost for each alternate access plan includes:

calculating the cost as a function of input/output cost and processing cost; and

calculating the input/output cost by scaling an estimated input/output cost by a scalar value associated with the percentage of the resource that is resident in working memory.

14. (Canceled).

15. (Original) The method of claim 3, wherein determining the retrieval status includes determining whether a beginning portion of the resource is resident in working memory, the method further comprising calculating a cost for the access plan based upon the determined retrieval status, including weighting the cost based upon the beginning portion of the resource being resident in working memory.

16. (Currently Amended) An apparatus, comprising:

at least one processor;

a memory that includes a working memory;

a resource manager; and

program code resident in the memory and configured to be executed by the at least one processor to optimize a database query by determining a retrieval status for a resource used by the database query, and generating an access plan for the database query using the determined retrieval status for the resource, wherein the program code is configured to determine the retrieval status by accessing the resource manager to obtain the retrieval status of the resource, wherein the resource manager is configured to track a percentage of the resource that is in the working memory, and wherein the resource manager is further configured to store the percentage of the resource that is in working memory in a header of a persistent copy of the resource.

17. (Original) The apparatus of claim 16, wherein the resource is selected from the group consisting of a database file, a database table, an index, a temporary result set, a temporary file, a hash table, and combinations thereof.

18. (Original) The apparatus of claim 16, wherein the program code is configured to determine the retrieval status by determining whether at least a portion of the resource is resident in the working memory.

19. (Original) The apparatus of claim 16, wherein the memory further includes a cache memory, and wherein the program code is configured to determine the retrieval status by determining whether at least a portion of the resource is resident in the cache memory.

20. (Original) The apparatus of claim 16, wherein the working memory includes a local memory, wherein the memory further includes a remote memory, and wherein the program code is configured to determine the retrieval status by determining whether at least a portion of the resource is resident in the local or the remote memory.

21. (Original) The apparatus of claim 16, wherein the program code is configured to determine the retrieval status by determining a percentage of the resource that is resident in the working memory.

22.-24. (Canceled).

25. (Original) The apparatus of claim 16, wherein the program code is configured to generate the access plan by generating a plurality of alternate access plans, calculating a cost for each alternate access plan using the determined retrieval status for the resource, and selecting one of the alternate access plans based upon the calculated costs for each alternate access plan.

26. (Original) The apparatus of claim 25, wherein the retrieval status for the resource indicates a percentage of the resource that is resident in the working memory, and wherein the program code is configured to calculate the cost for each alternate access plan by calculating the cost as a function of input/output cost and processing cost, and calculating the input/output cost by scaling an estimated input/output cost by a scalar value associated with the percentage of the resource that is resident in the working memory.

27. (Original) The apparatus of claim 25, wherein the program code is further configured to store the access plan and associate with the stored access plan a retrieval status assumption for the stored access plan, wherein the retrieval status assumption represents the determined retrieval status used to generate the access plan, and wherein the program code is further configured to, in response to a request to execute the stored access plan, determine a current retrieval status for the resource, compare the current retrieval status with the retrieval status assumption, and selectively generate another access plan for the database query using the current retrieval status based upon the comparison of the current retrieval status with the retrieval status assumption.

28. (Original) The apparatus of claim 16, wherein the program code is configured to determine the retrieval status by determining whether a beginning portion of the resource is resident in the working memory, and wherein the program code is further configured to calculate a cost for the access plan based upon the determined retrieval status, and weight the cost based upon the beginning portion of the resource being resident in the working memory.

29.-30. (Canceled).